VA Office of Information and Technology
Office of Enterprise Architecture Management

Systems Integration and Development Service

Configuration Item Definition Instruction CM313

Version 1.0

June 9, 2006

Department of Veterans Affairs

Systems and Integration Service

Letter of Promulgation

As the Director of the Systems Development and Integration Service (SIDS) within the Office of Enterprise Architecture at the Department of Veterans Affairs (VA), Office of Information and Technology (OI&T), I do hereby formally promulgate this Configuration Management Configuration Item (CI) Definition Instruction and approve its use for execution across the SIDS. This procedure provides direction and guidance to SIDS Program and Project personnel in the definition of CI in preparation for submittal of a CI for formal configuration control.

(Signature obtained and on file) 6/9/06
Frances G. Parker, Director (Acting) (Date)
Systems and Integration Service
Office of Enterprise Architecture Management
VA Office of Information and Technology

Record of Changes

CCP #	**CCP Date**	**Description of Change** (or title)	**Date Entered**	**Entered by:** (initials)
N/A	N/A	Initial issue of Version 1.0	6/9/06	bgl

Table of Contents

This Page Intentionally Blank

1. INTRODUCTION

This document has been prepared to address the "Configuration Item Definition" sub-section of Configuration Management Plans (CMP) within the Systems Integration and Development Service (SIDS) and as an aid to the SIDS Configuration Identification Procedure. This guide functions as direction and contains guidance to assist stakeholders in providing information necessary for properly placing configuration items (CI) under formal configuration control.

1.1. PURPOSE

The purpose of this document is to provide the descriptive data elements and attributes necessary for establishing formal configuration control over items designated as CI. Formal configuration control incorporates the establishment of baselines, the imposition of formal change management and change decision processes, the conduct of status accounting, and the oversight of auditing processes to ensure product integrity. The process of defining a CI includes identifying certain specific management and control information about it. The information is used to assist in the proper loading of files into automated configuration management (CM) applications and to ensure that only properly authorized persons and groups can make decisions about or incorporate changes to the CI. The completed Definition Template in Attachment 1 will be placed in the Dimensions automated CM application as a permanent part of the product configuration.

1.2. SCOPE

This procedure applies to all projects and systems within SIDS including the SIDS system of organizational management processes, plans, and procedures. Approval of a written Request for Waiver (RFW) to the Director, SIDS, is required for non-compliance with this document in whole or in part.

1.3. AUTHORITY

The Director, SIDS, is the issuing authority for this document and only the Director, SIDS, or higher authority may authorize it to be altered, superseded, or cancelled. Any changes or modifications to this document must be submitted for approval using the provisions of the SIDS CMP.

Any conflict between this document and higher authority will be resolved in favor of the higher authority. Anyone observing such a conflict is requested to bring it to the immediate attention of the Director, SIDS (or delegated SIDS authority).

1.4. CHANGE AND CANCELLATION

This is an original document and does not supersede a previous version or any other document.

1.5. REFERENCES

The following references were used in developing this procedure:

1) Carnegie-Mellon University/Software Engineering Institute, Capability Maturity Model Integration for Systems Engineering/Software/Integrated Products and Processes Development/Supplier Sources, March 2002.
2) Electronic Industries Alliance /Government Electronic and Information Technology Association Industries Alliance (EIA/GEIA) Standard-649-A, National Consensus Standard for Configuration Management, April 2004.

1.6. TERMS AND ABBREVIATIONS

1.6.1. Terms and Definitions

Configuration Item – Any product, product component, or product element selected for and placed under formal configuration management and control.
Configuration Component – Refer to "configuration item."
Configuration Element – Refer to "configuration item."
Configuration Unit – Refer to "configuration item."

NOTE: Configuration items can be decomposed into configuration components, units, and elements. These terms are often relative to the level of product organization at which they are envisioned. For example, an airplane (configuration item) may be decomposed into components – body, wings, tail section, engines, electronics, etc.; however, the builder of the engines may decompose the engine into its components. Therefore, the term "configuration item" in this document refers to any item, component, unit, or element selected for and placed under formal configuration management and control.

1.6.2. Abbreviations

The following abbreviations are used in this document:

Abbrev.	Expansion
CCP	Configuration Change Proposal
CI	Configuration Item
CM	Configuration Management
CML	Configuration Management Library
CMP	Configuration Management Plan
COTS	Commercial Off the Shelf
DBMS	Database Management System
EAM	(Office of) Enterprise Architecture Management
OI&T	Office of Information and Technology
RFW	Request For Waiver (and Deviation)
SDLC	Systems Development Life Cycle
SIDS	Systems Integration and Development Service
VA	(U.S. Department of) Veterans Affairs

1.7. ASSUMPTIONS

The definition of CIs assumes the following activities will occur as part of the Configuration Identification Selection Process:

- Selection of CIs at appropriate levels of breakdown within the product structure, and
- Assignments for project roles and responsibilities are documented.

2. PROCESS DIAGRAM

CI Definition is a single step in the CI Identification process and consists of completing the form in Attachment 1. Therefore, there is no process map for this document.

3. INSTRUCTION

3.1. DISCUSSION

After an item, component or element has been selected for formal configuration control, the provisions for maintaining the baseline must be established. These provisions define the baseline in terms of its descriptive and management attributes, i.e., those characteristics that facilitate governance of the item's configuration.

CI definition information serves several useful purposes. The information records the responsibility for and authority over each CI, and whether the CI is at the element level or is a grouping of elements. Each component item, as well as the product, might have a different release manager (database module/Senior Database Engineer, software module/Senior Software Engineer, User's Manual/Senior Editor, completed product/Program Manager). The information also makes these responsibility and authority assignments visible, so interested stakeholders can perform certain actions and know where to address questions and concerns.

Some data in the CI definition makes traceability possible between components and elements to help identify component interdependencies and cross-impacts. The traceability helps make it possible to correctly reassemble the components and elements into the parent or grandparent CI. The traceability information is also used for proper placement of an item within the structure of the CM Library (CML) for ease of access and proper control.

Finally, much of the CI definition information will be used in the version descriptions that accompany releases of products and product components.

Attachment 1 is a template for providing the necessary CI definition information. Section 3.2 provides instruction and guidance for completing the template.

3.2. REQUIRED DEFINITION INFORMATION

It is important to note that, although a project is a CI, its configuration is managed by managing the components, including internal and external interfaces, that comprise it. Therefore, the descriptive data is needed for each component and element of the CI. Due to differences in the functions of different elements, the type of information for each may differ, but there is equivalent information for each required information field. For example, "programming language" is required information for a software source code file; its equivalent for a product/project would be "operating system."

The following CI definition information is required only once for each item, component, or element file and that is when the first iteration is submitted to the automated CM application (Dimensions) for formal control. After the initial submittal, all necessary information will be automatically recorded by the CM application.

3.2.1. Filename

This is the name of the file to being forwarded to be entered into Dimensions. The name must comply with SIDS procedure, Configuration File Naming Conventions, which presents constraints and restrictions on file names. These constraints and restrictions are derived primarily from limitations presented by the commercial-off-the-shelf (COTS) management applications. The file name also must comply with any SIDS file naming convention for different types of files (database files, data management library files, source code files, scripts, engineering drawings, etc.).

3.2.2. Author

Author is the name of the person who can address technical questions about the item submitted to the CML, be it an internal working baseline or a released baseline. This may be the person who created the item or some other expert. The primary purpose of this information is to provide the best source of information during analysis of a Configuration Change Proposal (CCP) or during design and development of another component. After this first iteration of the item is placed under formal configuration control, analysts or the next "Change Agent" may have questions for the original writer. After submittal, the configuration management application will track all subsequent change transactions and authors.

3.2.3. Change Authority

Change Authority is the project role name or operational title with the authority for approving changes to the item. Do not use personal names. Acceptable entries can be found in the project management plan and other plans ancillary to it, such as the CM Plan, Quality Assurance Plan, Requirements Management Plan, etc.

3.2.4. Change Agent

Change Agent is the specific project role or group name or organizational title with the authority to check a file out of the CML, modify it, and check it back in. Examples include: Design Engineer (for design drawings), Software Engineer (for source code changes), Tech Writer (for document changes), etc. As can be seen, the Change Agent indicated may be a title that is occupied by a single person or the name of a group that is comprised of several individuals. (In practice, it would be up to the responsible manager to assign the incorporation duty for each change that had to be performed during the course of the project. The responsible manager is usually designated in a project management plan.)

3.2.5. Release Authority

Release Authority is the individual or specific review team (usually represented by the chair) who may authorize the release (or publication) of the file outside the group responsible for its development. The Release Authority may be the same for every product piece produced (very small product), or it may change from one product life cycle stage to the next (large, long-term products), by component type (e.g., one for specification documents, one for designs, one for database components, and one for software components), or one for each individual component.

3.2.6. Parent Product/Project Group

Parent Project Group (or Parent Product Group) is the highest level name of the project or product. If the product group is comprised of several projects, enter both the overall product name and the name of the component project. For example, Requirements and Eligibility, Contact Management, and Veteran's Information Service are all component project of the envisioned OneVA product. Therefore, "OneVA/Contact Management" would be an appropriate entry for a file in the Contact Management set of work products.

3.2.7. Parent item

Parent Item refers to the next "senior component" that was decomposed to yield this item. This information assists the CM facilitator in determining the proper placement of the file within the CML. Additionally, this information assists in the subsequent traceability of the configuration components and elements within a configuration item.

3.2.8. Purpose

The Purpose identifies in general terms what the file is used for. Be brief and high-level in the purpose description. Often, this may be a repeat of a developer's remarks line within the file. The following hypothetical examples are intended to provide a flavor of what is required:

"Create finance tables."
"Create personal address tables."
"Call up the user's log-in screen."
"Initiate the User Manual."

3.2.9. File Type

File Type identifies the "format" of the file; for example: MS Word XP Document, Source Code, Visio 7.0 Drawing, ERWin Design, Script, Data Definition Language, etc. This information helps to identify the tools needed for its continued maintenance. If a tool is named, the name should include the version of the tool used.

3.2.10. Language, OS, DBMS

The Language, OS, and DBMS field helps to place the item within the correct variant group of a project and to identify the specific capabilities (personal or environmental) needed to read and maintain the file contents. For example, a particular product may have a variant built to run in an Oracle 8i environment, one for an Oracle 9i environment, and yet another in a Sybase environment.

Enter the information as applicable, i.e., programming language for a source code file, operating system for the resulting executable, and database management system (DBMS) for database components. Product manuals such as a User Manual or Installation Manual should be labeled as "All variants" unless unique variants of the manual are required. For management documents, indicate the organizational name (or specific project name) to which the document belongs.

3.2.11. Interfaces and Interdependent Items

In this area, list all known interface components and all known interdependent items. An interface is a distinct product component that is integral between two or more other components. An example would be a wing-mount on an airplane (an interface between the wing and the body). Such a wing-mount not only holds the wing on, but also provides an avenue for fuel lines, hydraulic lines, and perhaps data feedback lines from computer components in the wing to the flight recorder. Two database tables that use each other's data to derive additional data are interdependent; i.e., each has autonomous control over the fields and formulas within it, but both must be coordinated together to ensure compatibility. A project CM plan has an interdependency with the organization structure of the project management plan for CM roles and responsibilities assignments.

ATTACHMENT 1 – CI DEFINITION TEMPLATE

The following attributes comprise the definition of this CI:

1. **Filename**

2. **Author**

3. **Change Authority**

4. **Change Agent**

5. **Release Authority**

6. **Parent Product/Project Group**

7. **Parent Item**

8. **Purpose**

9. **File type (e.g., Visio drawing, ERWin design, MS Word XP document, etc.)**

10. **Language, OS, DBMS**

11. **Interface (y/n) and interdependent items**

Note: This document is required only upon the first submittal of a selected CI to formal configuration control.